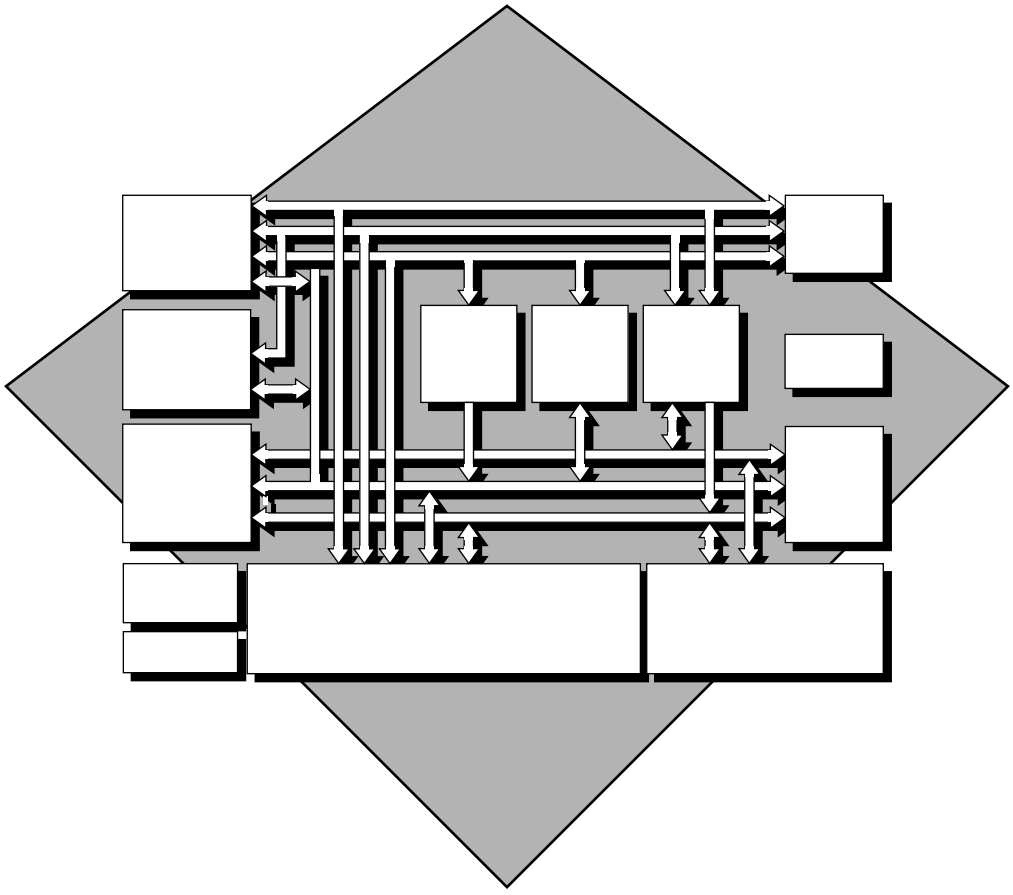
SECTION 1

DSP56602 OVERVIEW



1.1 INTRODUCTION

This manual describes the DSP56602 16-bit Digital Signal Processor (DSP), its memory and operating modes, and its peripheral modules. This manual is intended to be used with the *DSP56600 Family Manual (DSP56600FM/AD),* which describes the Central Processing Unit (CPU), programming models, and instruction set details. The *DSP56602 Technical Data Sheet (DSP56602/D)* provides electrical specifications, timing, pinout, and packaging descriptions on the DSP56602. These documents, as well as Motorola's DSP development tools, can be obtained through a local Motorola Semiconductor Sales Office or authorized distributor.

To receive the latest information, access the Motorola DSP home page located at the address listed on the back cover of this document.

The DSP56602 is a member of the DSP56600 core-based family of programmable CMOS DSPs. This general purpose DSP combines processing power with configuration flexibility, making it an excellent cost-effective solution for signal processing and control functions.

This manual is arranged in the following sections:

- **Section 1—DSP56602 Overview** provides a brief overview of the DSP56602, describes the structure of this document, and lists other documentation necessary to use this chip.
- **Section 2—Signal/Connection Description** provides a description of the signals present on the pins of the DSP56602, and how these signals are grouped into the various interfaces.
- **Section 3—Memory Maps** describes the on-chip memory, structures, registers, and interfaces.
- **Section 4—Core Configuration** describes the registers that must be programmed to properly configure the DSP56600 core when using the DSP56602.
- **Section 5—External Memory Interface (Port A)** describes the External Memory Interface, which is also referred to as Port A.
- **Section 6—GPIO** describes the dedicated General Purpose Input/Output (GPIO) interface, and the alternate GPIO functionality provided on certain on-chip interfaces.
- **Section 7— Host Interface (HI08)** describes the 8-bit HI08 Host Interface.
- **Section 8—Synchronous Serial Interface** describes the 16-bit Synchronous Serial Interface (SSI), which communicates with devices such as codecs, other DSPs,

microprocessors, and peripherals, to provide the primary data input path. The SSI is a part of Port C.

- **Section 9—Triple Timer Module** describes the three internal timer/counter devices.
- **Section 10—On-Chip Emulation Module** describes the On-Chip Emulation (OnCE™) module, which is accessed through the Joint Test Action Group (JTAG) port.
- **Section 11—JTAG Port** describes the specifics of the JTAG port on the DSP56602.
- **Appendix A—Bootstrap Program** provides a sample listing of bootstrap code, intended for use as an example of the bootstrap code than can be developed by the customer to start or reset the DSP56602.
- **Appendix B—X I/O Equates** lists the Input/Output equates for the DSP56602.
- **Appendix C—BSDL Listing** provides the Boundary Scan Description Listing (BSDL) for the DSP56602.
- **Appendix D—Programmer's Reference** provides programming references and master programming sheets used to program the DSP56602 registers.
- **Index** provides a cross-reference to topics in this manual.

1.2 MANUAL CONVENTIONS

The following conventions are used in this manual:

- Bits within registers are always listed from Most Significant Bit (MSB) to Least Significant Bit (LSB).

Note: Other manuals may use the opposite convention, with bits listed from LSB to MSB.

- Bits within a register are indicated AA[n:0] when more than one bit is involved in a description. For purposes of description, the bits are presented as if they are contiguous within a register. However, this is not always the case. Refer to the programming model diagrams or to the programmer's sheets to see the exact location of bits within a register.
- When a bit is described as "set," its value is 1. When a bit is described as "cleared," its value is 0.
- Pins or signals that are asserted low (made active when pulled to ground) have an overbar over their name; for example, the $\overline{SS0}$ pin is asserted low.

- Hex values are indicated with a dollar sign ($) preceding the hex value as follows: $FFFF is the X memory address for the Interrupt Priority Register—Core (IPR-C).
- Code examples are displayed in a monospaced font, as shown in **Example 1-1**.

Example 1-1 Sample Code Listing

```
BFSET  #$0007,X:PCC; Configure:                          line 1
                    ; MISO0, MOSI0, SCK0 for SPI master  line 2
                    ; ~SS0 as PC3 for GPIO               line 3
```

- Pins or signals listed in code examples that are asserted low have a tilde in front of their names. In the previous example, line 3 refers to the $\overline{SS0}$ pin (shown as ~SS0).
- The word “assert” means that a high true (active high) signal is pulled high to V_{CC} or that a low true (active low) signal is pulled low to ground. The word “deassert” means that a high true signal is pulled low to ground or that a low true signal is pulled high to V_{CC}. See **Table 1-1**.

Table 1-1 High True / Low True Signal Conventions

Signal/Symbol	Logic State	Signal State	Voltage
$\overline{PIN}$	True	Asserted	Ground[1]
$\overline{PIN}$	False	Deasserted	V_{CC}[2]
PIN	True	Asserted	V_{CC}
PIN	False	Deasserted	Ground

Notes: 1. Ground is an acceptable low voltage level. See the appropriate data sheet for the range of acceptable low voltage levels (typically a TTL logic low).
2. V_{CC} is an acceptable high voltage level. See the appropriate data sheet for the range of acceptable high voltage levels (typically a TTL logic high).

- The word “reset” is used in four different contexts in this manual:
  - There is a reset pin that is always written as “$\overline{RESET}$”. The word “pin” is a generic term for any pin on the chip.
  - There is a reset instruction that is always written as “RESET”
  - The word reset refers to the reset function and is written in lower case with a leading capital letter as grammar dictates
  - “Reset” refers to the Reset state.

1.3 DSP56600 CORE DESCRIPTION

The DSP56600 core is based on the DSP56300 core, with a number of power-saving, performance-enhancing, and cost-reducing features implemented. With its seven-stage instruction pipeline, the DSP56600 core is capable of executing an instruction on every clock cycle. A standard interface between the DSP56600 core and the on-chip memory and peripherals supports many memory and peripheral configurations. Complete details of the DSP56600 core are provided in the *DSP56600 Family Manual (DSP56600FM/AD)*.

The following are some of the features of the DSP56600 core:

- 60 Million Instructions Per Second (MIPS) with a 60 MHz clock at 2.7 V
- Fully pipelined 16 × 16-bit parallel Multiplier-Accumulator (MAC)
- 40-bit parallel barrel shifter
- Highly parallel instruction set with unique DSP addressing modes
- Code compatible with the 56300 core
- Position Independent Code (PIC) support
- Nested hardware DO loops
- Fast auto-return interrupts
- On-chip support for software patching and enhancements
- On-chip Phase Lock Loop (PLL)
- Real-time trace capability via External Address Bus
- On-Chip Emulation (OnCE) module
- JTAG port

1.4 DSP56600 CORE FUNCTIONAL BLOCKS

The DSP56600 core provides the following functional blocks:

- Data Arithmetic Logic Unit (Data ALU)
- Address Generation Unit (AGU)
- Program Control Unit (PCU)
- Program Patch Logic

- PLL and Clock Oscillator
- Expansion Memory Interface (Port A)
- JTAG Test Access Port and On-Chip Emulation (OnCE) module
- Memory

In addition, the DSP56602 provides a set of on-chip peripherals, described in **DSP56602 Architecture Overview** on page 1-13.

1.4.1 Data Arithmetic Logic Unit (ALU)

The Data Arithmetic Logic Unit (ALU) performs all the arithmetic and logical operations on data operands in the DSP56600 core. The components of the Data ALU are as follows:

- Four 16-bit input general purpose registers: X1, X0, Y1, and Y0
- A parallel, fully pipelined Multiplier-Accumulator unit (MAC)
- Six Data ALU registers (A2, A1, A0, B2, B1, and B0) that are concatenated into two general purpose, 40-bit accumulators, A and B
- An accumulator shifter that is an asynchronous parallel shifter with a 40-bit input and a 40-bit output
- A Bit Field Unit (BFU) with a 40-bit barrel shifter
- Two data bus shifter / limiter circuits

1.4.1.1 Data ALU Registers

The Data ALU registers can be read or written over the X Data Bus (XDB) and the Y Data Bus (YDB) as 16- or 32-bit operands. The source operands for the Data ALU, which can be 16, 32, or 40 bits, always originate from Data ALU registers. The results of all Data ALU operations are stored in an accumulator.

All the Data ALU operations are performed in 2 clock cycles in pipeline fashion so that a new instruction can be initiated in every clock, yielding an effective execution rate of one instruction per clock cycle. The destination of every arithmetic operation can be used as a source operand for the immediate following operation without penalty.

1.4.1.2 Multiplier-Accumulator (MAC)

The Multiplier-Accumulator (MAC) unit comprises the main arithmetic processing unit of the DSP56600 core and performs all of the calculations on data operands. In the case of arithmetic instructions, the unit accepts as many as three input operands and outputs

one 40-bit result of the following form, Extension:Most Significant Product:Least Significant Product (EXT:MSP:LSP).

The multiplier executes 16-bit × 16-bit, parallel, fractional multiplies, between two's-complement signed, unsigned, or mixed operands. The 32-bit product is right-justified and added to the 40-bit contents of either the A or B accumulator. A 40-bit result can be stored as a 16-bit operand. The LSP can either be truncated or rounded into the MSP. Rounding is performed if specified.

1.4.2 Address Generation Unit

The AGU performs the effective address calculations using integer arithmetic necessary to address data operands in memory and contains the registers used to generate the addresses. It implements four types of arithmetic: linear, modulo, multiple wrap-around modulo, and reverse-carry. The AGU operates in parallel with other chip resources to minimize address-generation overhead.

The AGU is divided into two halves, each with its own Address Arithmetic Logic Unit (Address ALU). Each Address ALU has four sets of register triplets, and each register triplet is composed of an address register, an offset register, and a modifier register. The two Address ALUs are identical. Each contains a 16-bit full adder (called an offset adder).

A second full adder (called a modulo adder) adds the summed result of the first full adder to a modulo value that is stored in its respective modifier register. A third full adder (called a reverse-carry adder) is also provided.

The offset adder and the reverse-carry adder are in parallel and share common inputs. The only difference between them is that the carry propagates in opposite directions. Test logic determines which of the three summed results of the full adders is output.

Each Address ALU can update one address register from its respective address register file during 1 instruction cycle. The contents of the associated modifier register specifies the type of arithmetic to be used in the address register update calculation. The modifier value is decoded in the Address ALU.

1.4.3 Program Control Unit

The Program Control Unit (PCU) performs instruction prefetch, instruction decoding, hardware DO loop control, and exception processing. The PCU implements a

seven-stage pipeline and controls the different processing states of the DSP56600 core. The PCU consists of three hardware blocks:

- Program Decode Controller (PDC)
- Program Address Generator (PAG)
- Program Interrupt Controller (PIC)

The PDC decodes the 24-bit instruction loaded into the instruction latch and generates all signals necessary for pipeline control. The PAG contains all the hardware needed for program address generation, system stack, and loop control. The PIC arbitrates among all interrupt requests (internal interrupts, as well as the five external requests $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, $\overline{\text{IRQC}}$, $\overline{\text{IRQD}}$, and $\overline{\text{NMI}}$), and generates the appropriate interrupt vector address.

The PCU implements its functions using the following registers:

- PC—Program Counter register
- SR—Status Register
- LA—Loop Address register
- LC—Loop Counter register
- VBA—Vector Base Address register
- SZ—Size register
- SP—Stack Pointer
- OMR—Operating Mode Register
- SC—Stack Counter register

The PCU also includes a hardware System Stack (SS).

1.4.4 Program Patch Logic

The Program Patch Logic (PPL) block provides the DSP56600 core user a way to fix the program code in the on-chip ROM without generating a new mask. Implementing the code correction is done by replacing a piece of ROM-based code with a patch program stored in RAM. The PPL consists of four Patch Address Registers (PAR1–PAR4) and four patch address comparators. Each PAR points to a starting location in the ROM code where the program flow is to be changed. The PC register in the PCU is compared to each PAR. When an address of a fetched instruction is identical to an address stored in one of the PARs, the Program Data Bus (PDB) is forced to a corresponding JMP

instruction, replacing the instruction that otherwise would have been fetched from the ROM.

1.4.5 PLL and Clock Oscillator

The DSP56600 core features a Phase Lock Loop (PLL) clock oscillator in its central processing module. The PLL allows the processor to operate at a high internal clock frequency using a low frequency clock input, a feature that offers two immediate benefits:

- A lower frequency clock input reduces the overall electromagnetic interference generated by a system.
- The ability to oscillate at different frequencies reduces costs by eliminating the need to add additional oscillators to a system.

The clock generator in the DSP56600 core is composed of two main blocks: the PLL, which performs clock input division, frequency multiplication, and skew elimination; and the Clock Generator (CLKGEN), which performs low power division and clock pulse generation.

1.4.6 Expansion Memory Interface (Port A)

Port A is the memory expansion port used for both program and data memory. It provides an easy to use, low part-count connection with fast or slow static memories and with I/O devices. The Port A data bus is 24 bits wide with a separate 16-bit address bus capable of a sustained rate of one memory access per two clock cycles. External memory can be as large as 64 K × 24-bit program memory space, depending on chip configuration. An internal wait state generator can be programmed to insert as many as thirty-one wait states if access to slower memory or I/O device is required. For power-sensitive applications and applications that do not require external memory, Port A can be fully disabled.

1.4.7 JTAG Test Access Port and On-Chip Emulation Module

The DSP56600 core provides a dedicated user-accessible Test Access Port (TAP) that is fully compatible with the *IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture*. Problems associated with testing high density circuit boards have led to development of this standard under the sponsorship of the Test Technology Committee

of IEEE and the Joint Test Action Group (JTAG). The DSP56600 core implementation supports circuit-board test strategies based on this standard.

The test logic includes a TAP consisting of four dedicated signal pins, a 16-state controller, and three test data registers. A Boundary Scan Register links all device signal pins into a single shift register. The test logic, implemented utilizing static logic design, is independent of the device system logic. More information on the JTAG port is provided in **Section 11, JTAG Port.**

The On-Chip Emulation (OnCE) module provides a means of interacting with the DSP56600 core and its peripherals non-intrusively so that a user can examine registers, memory, or on-chip peripherals. This facilitates hardware and software development on the DSP56600 core processor. OnCE module functions are provided through the JTAG TAP pins. More information on the OnCE module is provided in **Section 10, On-Chip Emulation Module.**

1.4.8 On-Chip Memory

The memory space of the DSP56600 core is partitioned into program memory space, X data memory space, and Y data memory space. The data memory space is divided into X data memory and to Y data memory in order to work with the two Address ALUs and to feed two operands simultaneously to the Data ALU. Memory space includes internal RAM and ROM and can be expanded off-chip under software control. More information on the internal memory is provided in **Section 3, Memory Maps.**

1.5 INTERNAL BUSES

To provide data exchange between these blocks, the following buses are implemented:

- Peripheral I/O Expansion Bus (PIO_EB) to peripherals
- Program Memory Expansion Bus (PM_EB) to Program ROM
- X Memory Expansion Bus (XM_EB) to X Memory
- Y Memory Expansion Bus (YM_EB) to Y Memory
- Global Data Bus (GDB) between Program Control Unit and other core structures
- Program Data Bus (PDB) for carrying program data throughout the core
- X Memory Data Bus (XDB) for carrying X data throughout the core

- Y Memory Data Bus (YDB) for carrying Y data throughout the core
- Program Address Bus (PAB) for carrying program memory addresses throughout the core
- X Memory Address Bus (XAB) for carrying X memory addresses throughout the core
- Y Memory Address Bus (YAB) for carrying Y memory addresses throughout the core

With the exception of the Program Data Bus (PDB), all internal buses on the DSP56600 family members are 16-bit buses. The PDB is a 24-bit bus. **Figure 1-1** provides a block diagram of the DSP56602.

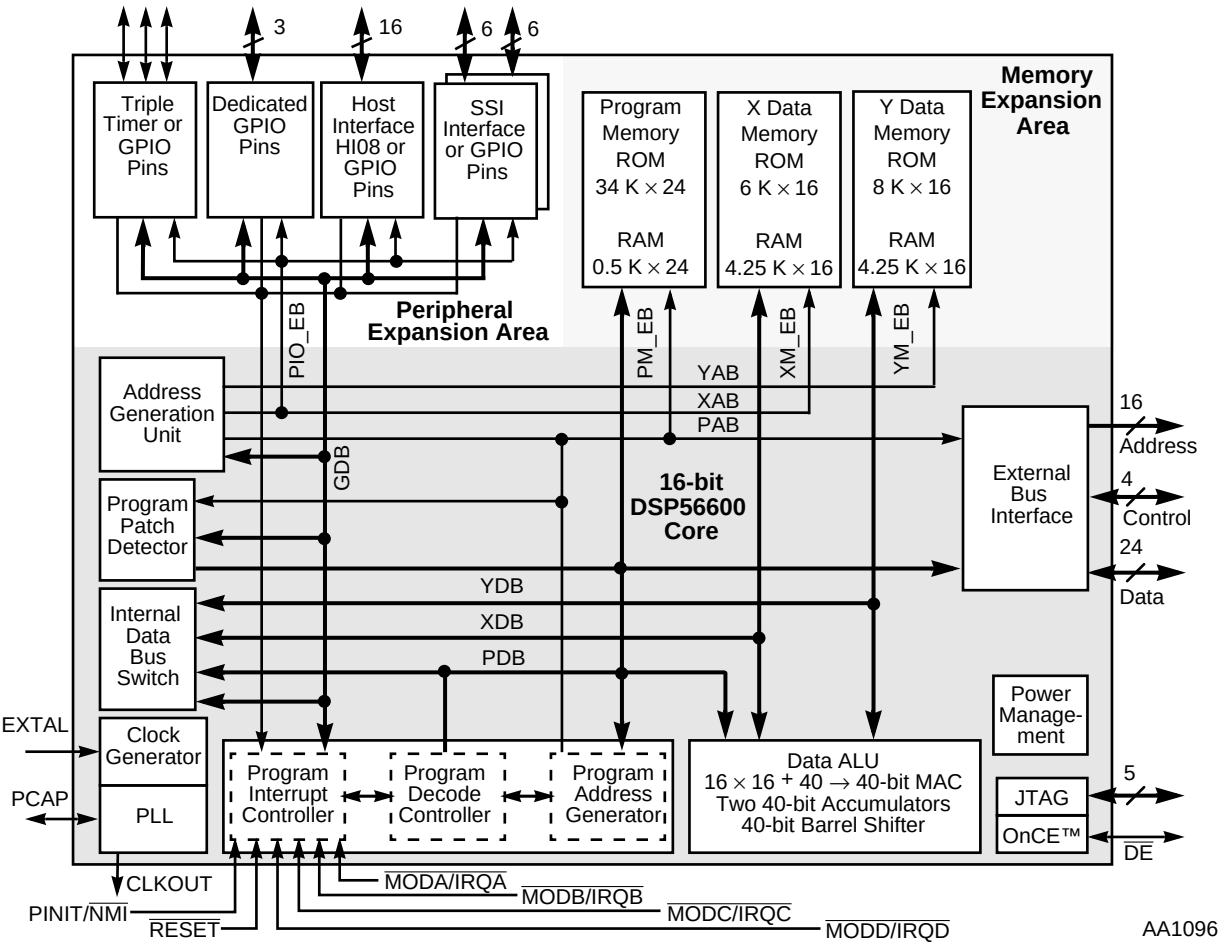


Figure 1-1 DSP56602 Block Diagram

1.6 DSP56602 ARCHITECTURE OVERVIEW

The DSP56602 is designed to perform a wide variety of fixed-point digital signal processing functions. In addition to the core features previously discussed, the DSP56602 provides the following peripherals:

- Three dedicated General Purpose I/O (GPIO) pins
- As many as thirty-one additional user-configurable GPIO pins
- 8-bit parallel Host Interface (HI08) to external hosts
- Dual Synchronous Serial Interface (SSI)
- Triple timer module
- Four external interrupt/mode control lines

1.6.1 GPIO Functionality

The General Purpose I/O (GPIO) port consists of three bidirectional pins, each pin separately controlled. Functionality is controlled by three memory-mapped registers. GPIO functionality is also available on the HI08, SSI, and timer pins when these pins are not otherwise being used by their peripherals. The techniques for register programming for all GPIO functionality is very similar between these interfaces. A maximum of thirty-four GPIO pins can be configured.

1.6.2 Host Interface (HI08)

The Host Interface (HI08) is a byte-wide, full-duplex, double-buffered, parallel port that can be connected directly to the data bus of a host processor. The HI08 supports a variety of buses, and provides connection with a number of industry-standard DSPs, microcomputers, and microprocessors without requiring any additional logic.

The DSP core views the HI08 as a memory-mapped peripheral occupying eight 16-bit words in data memory space. The DSP can use the HI08 as a memory-mapped peripheral, using either standard polled or interrupt programming techniques. Separate transmit and receive data registers are double-buffered to allow the DSP and host processor to efficiently transfer data at high speed. Memory mapping allows DSP core communication with the HI08 registers to be accomplished using standard instructions and addressing modes.

1.6.3 Synchronous Serial Interface (SSI)

The DSP56602 provides two independent and identical Synchronous Serial Interfaces (SSIs). Each SSI provides a full-duplex serial port for communication with a variety of serial devices, including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals that implement the Motorola SPI. The SSI consists of independent transmitter and receiver sections and a common SSI clock generator.

The capabilities of the SSI include:

- Independent (asynchronous) or shared (synchronous) transmit and receive sections with separate or shared internal / external clocks and frame syncs
- Normal mode operation using frame sync
- Network mode operation with as many as 32 time slots
- Programmable word length (8, 12, or 16 bits)
- Program options for frame synchronization and clock generation

1.6.4 Triple Timer

The triple timer module is composed of a common 14-bit prescaler and three independent and identical general purpose 16-bit timer / event counters, each one having its own memory-mapped register set.

Each timer can use internal or external clocking and can interrupt the DSP after a specified number of events (clocks) or can signal an external device after counting internal events. Each timer connects to the external world through one bidirectional pin. When this pin is configured as an input, the timer can function as an external event counter or measures external pulse width / signal period. When the pin is used as an output, the timer can function as either a timer, a watchdog, or a Pulse Width Modulator (PWM).